Bellabeat Case Study

Adegbite Moriam Adesola

10/10/2022

INTRODUCTION

Bellabeat is a high-tech company founded in 2013 by Urška Sršen and Sando Mur. They are a manufacturer of health-focused smart products for women. By collecting data on activity, sleep, stress and reproductive health, Bellabeat empowered women with knowledge about their health and habits. Bellabeat is a successful small company that has grown rapidly and quickly positioned itself as a tech-driven wellness company for women, but they have the potential to become a larger player in the global smart device market.

As a junior data analyst working on the marketing analyst team at Bellabeat, I've been asked to focus on a Bellabeat product and analyze smart device usage data to gain insight into how people are already using their smart devices. The insights derived would be used to provide high-level recommendations that will guide Bellabeat's marketing strategy.

Bellabeat Products

- **Bellabeat app:** The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products.
- **Leaf:** Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress.
- **Time:** This wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress. The Time watch connects to the Bellabeat app to provide you with insights into your daily wellness.
- **Spring:** This is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels.
- **Bellabeat membership:** Bellabeat also offers a subscription-based membership program for users. Membership gives users 24/7 access to fully personalized guidance on nutrition, activity, sleep, health and beauty, and mindfulness based on their lifestyle and goals.

ASK

The **business task** is to analyze smart device usage data to gain insights into how consumers use non-Bellabeat smart devices. Then, apply these insights to one Bellabeat product in my presentation and give recommendations on how the trends can help Bellabeat marketing strategy.

Guiding questions for analysis
1. What are some trends in smart device usage?

2. How could these trends apply to Bellabeat customers?
3. How could these trends help influence Bellabeat marketing strategy?

Key stakeholders
* **Urška Sršen:** Bellabeat's cofounder and Chief Creative Officer
* **Sando Mur:** Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team
* **Bellabeat marketing analytics team:** A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

PREPARE

The public data which explored smart device users' daily habits used for this analysis is Fitbit Fitness Tracker Data. The data was made available through Mobius on Kaggle. The dataset contains personal tracker fitness data of thirty Fitbit users collected via a consented distributed survey via Amazon Mechanical Turk between 03-12-2016 and 05-12-2016. The data includes information about daily activity, steps, heart rate and sleep monitoring. Individual reports can be parsed by export session ID (column A) or timestamp (column B). Variation between outputs represents the use of different types of Fitbit trackers and individual tracking behaviors / preferences. The dataset contained 18 CSV documents including minute-level output for physical activity, heart rate, and sleep monitoring which are stored in long formats. The limitations of the data are its sample size and the absence of descriptions of the users, such as age and gender.

PROCESS

Loading packages and setting up work environment

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
setwd("~/Data/Google Data Analytics/Bellabeat")
```

Importing datasets

Three datasets were used for analysis. They include the data on daily activity, sleep and weight.

```
dailyActivity <- read.csv("dailyActivity_merged.csv")
sleepDay <- read.csv("sleepDay_merged.csv")
weightLog <- read.csv("weightLogInfo_merged.csv")
```

Inspecting and cleaning data

The `str() function` was used to return information about the internal structure of the data, including information on the number of columns (variables) and rows (observations), the names of the column, the data type and a few observations.

Daily Activity

```
str(dailyActivity)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

Sleep

```
str(sleepDay)
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM"
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

Weight

```
str(weightLog)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "4/21/20
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

The `n_distinct() function` was used to determine the number of unique user IDs in the three datasets.

```
n_distinct(dailyActivity$Id)
```

```
## [1] 33
```

```
n_distinct(sleepDay$Id)
```

```
## [1] 24
```

```
n_distinct(weightLog$Id)
```

```
## [1] 8
```

There are 33, 24, and 8 unique user IDs in the daily activity, sleep and weight data, respectively. Although the sleep and weight data has insufficient records, they will be used in the analyze stage to identify potential areas of improvement.

Changing of data type

The date column in `dailyActivity` and `sleepDay` datasets will be formatted from character data type to date, as they will be used later in the analysis.

```
dailyActivity <- dailyActivity %>%
  mutate(ActivityDate = as.Date(ActivityDate, format = "%m/%d/%Y"))
str(dailyActivity)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : Date, format: "2016-04-12" "2016-04-13" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
sleepDay <- sleepDay %>%
  mutate(SleepDate = as.Date(SleepDay, format = "%m/%d/%Y"))
str(sleepDay)
```

```
## ’data.frame’:    413 obs. of  6 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM"
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
##  $ SleepDate         : Date, format: "2016-04-12" "2016-04-13" ...
```

ANALYZE AND SHARE

Data summaries

Daily Activity

```
dailyActivity %>%
  select(TotalSteps,TotalDistance,VeryActiveMinutes,FairlyActiveMinutes,
         LightlyActiveMinutes,SedentaryMinutes,Calories) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    VeryActiveMinutes FairlyActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :  0.00    Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:  0.00    1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :  4.00    Median :  6.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 21.16    Mean   : 13.56
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 32.00    3rd Qu.: 19.00
##  Max.   :36019   Max.   :28.030   Max.   :210.00    Max.   :143.00
##  LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.0        Min.   :   0.0   Min.   :   0
##  1st Qu.:127.0        1st Qu.: 729.8   1st Qu.:1828
##  Median :199.0        Median :1057.5   Median :2134
##  Mean   :192.8        Mean   : 991.2   Mean   :2304
##  3rd Qu.:264.0        3rd Qu.:1229.5   3rd Qu.:2793
##  Max.   :518.0        Max.   :1440.0   Max.   :4900
```

The summary shows that the maximum sedentary minutes is 1440 minutes, which is the total minutes in a day. Therefore, I assumed that the sedentary minutes column include the time asleep.

Sleep

```
sleepDay %>%
  select(TotalSleepRecords,TotalMinutesAsleep,TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

Weight

```
weightLog %>%
  select(WeightKg,Fat,BMI) %>%
  summary()
```

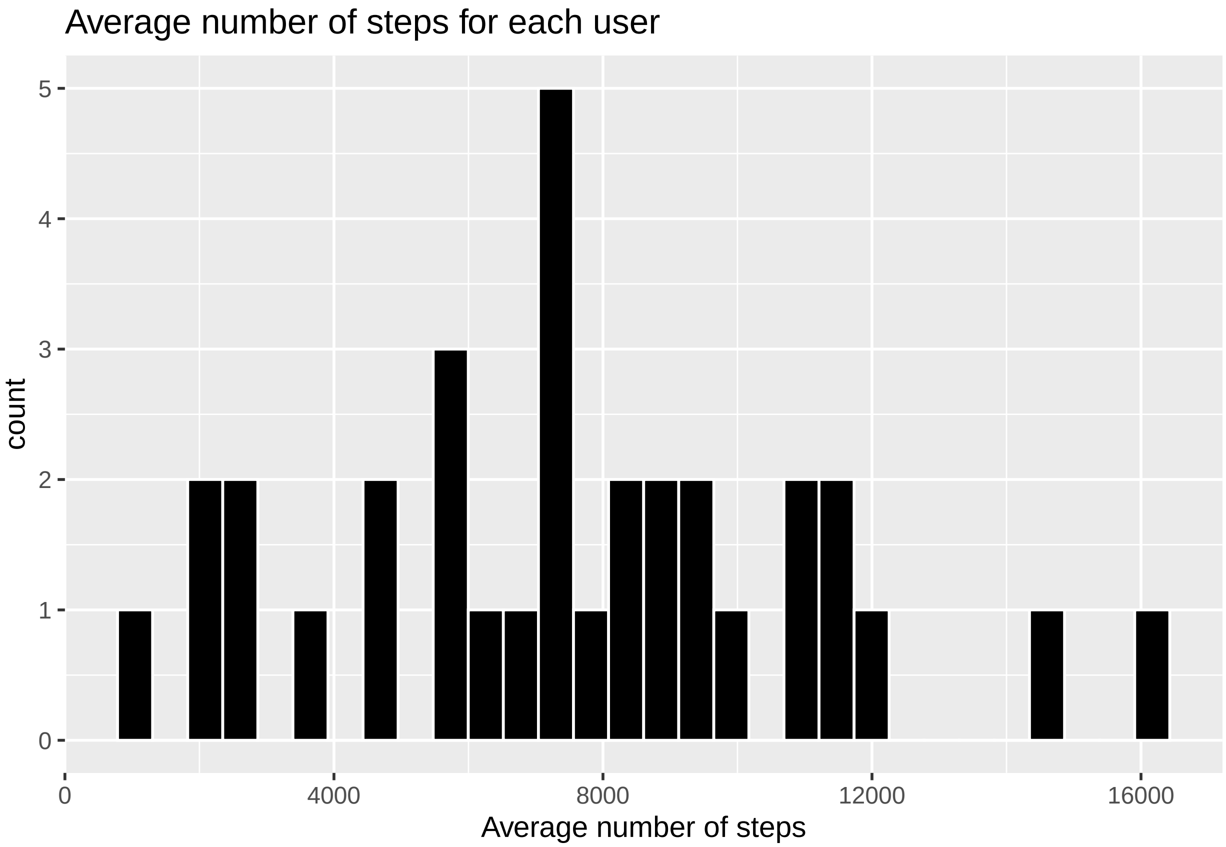
```
##     WeightKg           Fat             BMI
##  Min.   : 52.60   Min.   :22.00   Min.   :21.45
##  1st Qu.: 61.40   1st Qu.:22.75   1st Qu.:23.96
##  Median : 62.50   Median :23.50   Median :24.39
##  Mean   : 72.04   Mean   :23.50   Mean   :25.19
##  3rd Qu.: 85.05   3rd Qu.:24.25   3rd Qu.:25.56
##  Max.   :133.50   Max.   :25.00   Max.   :47.54
##                   NA's   :65
```

Histogram plot of the average number of steps for each user

```
userAvgDailySteps <- dailyActivity %>% group_by(Id) %>%
  summarise(AverageDailySteps = mean(TotalSteps))
```

```
ggplot(data=userAvgDailySteps)+
  geom_histogram(mapping = aes(x=AverageDailySteps), color='white', fill='black')+
  labs(title = 'Average number of steps for each user',
       x = 'Average number of steps')
```

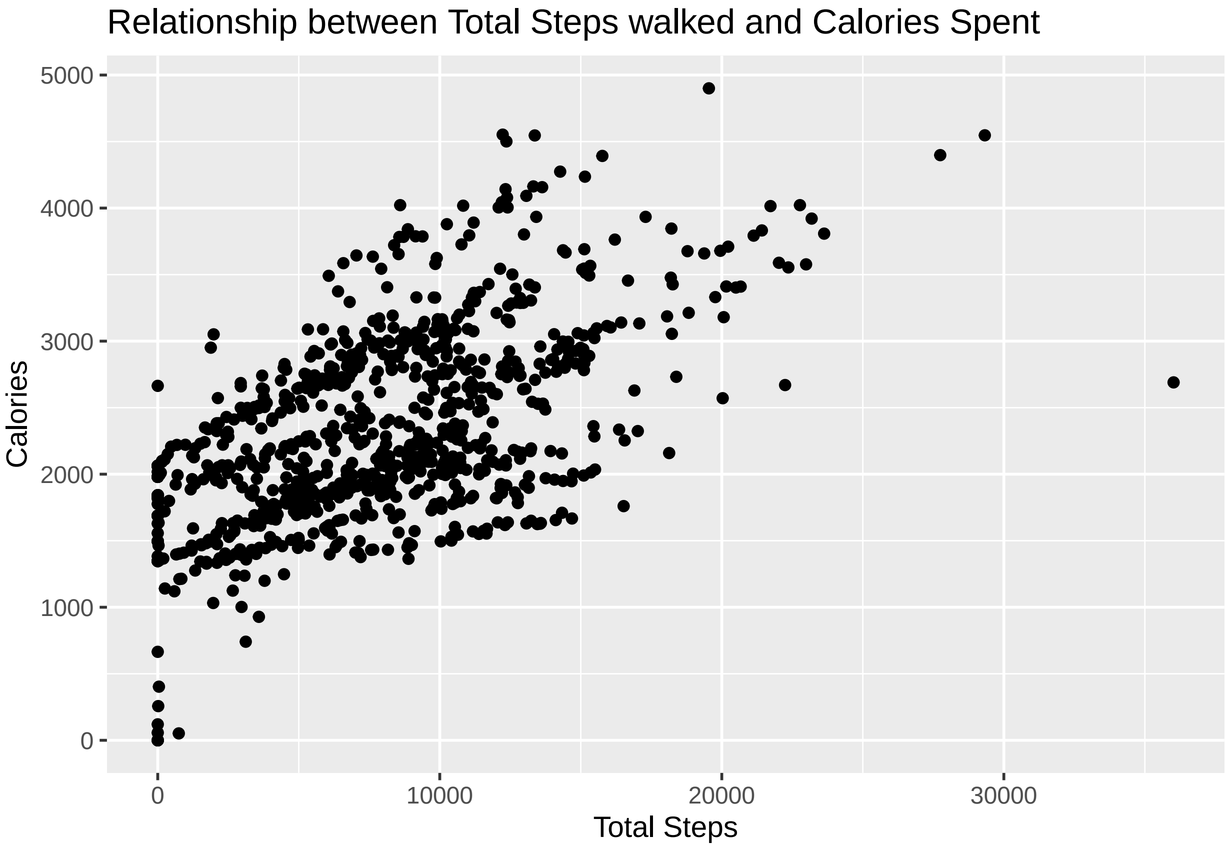
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The plot shows that six users walked an average of less than 4000 steps while seven users walked an average of more than 10000 steps. A total of 10000 steps daily is recommended.

Relationship between total steps and calories spent

```
ggplot(data=dailyActivity)+
  geom_point(mapping = aes(x=TotalSteps, y=Calories))+
  labs(title = 'Relationship between Total Steps walked and Calories Spent',
       x = 'Total Steps',
       y = 'Calories')
```



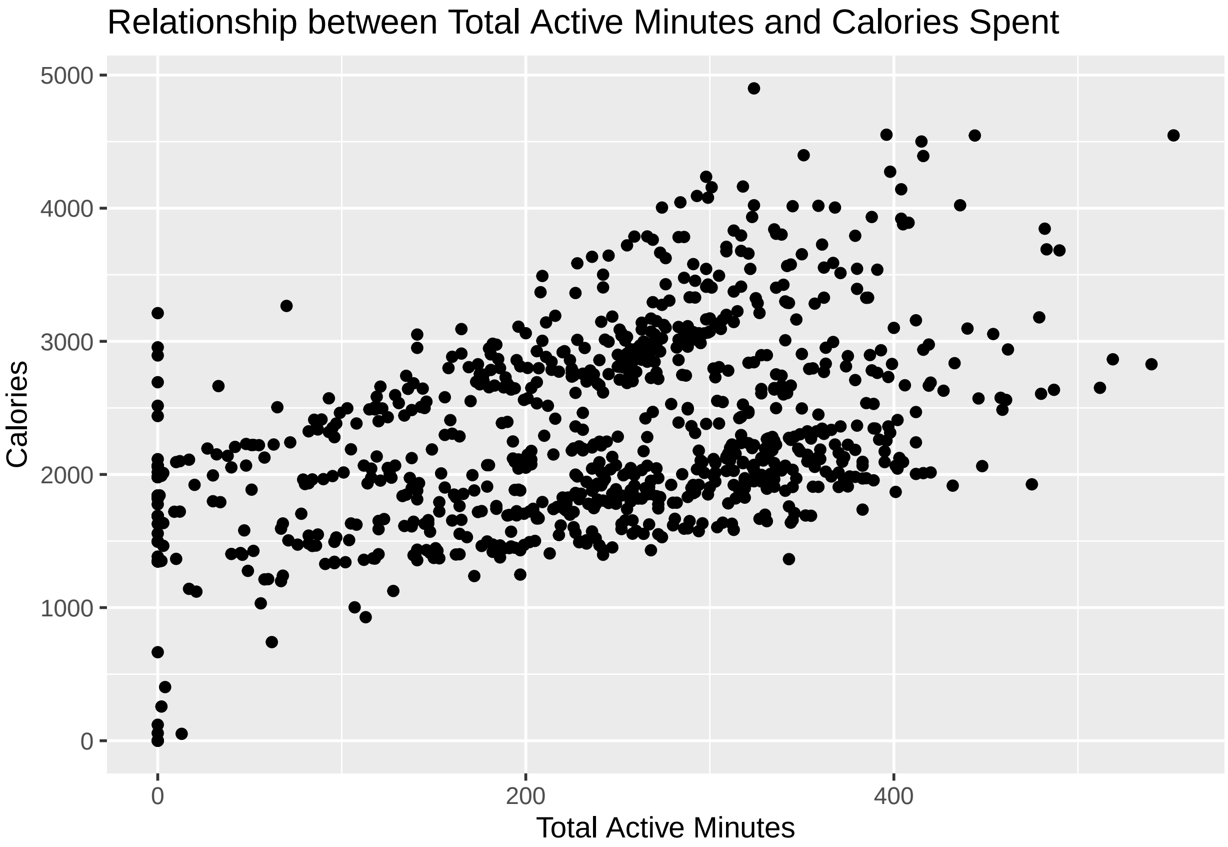
The plot shows a positive relationship between total steps and calories. As the number of steps increased, the calories spent increased. However, there are records of zero step with positive values for calories spent.

Relationship between active minutes and calories

The total active minutes is the sum of the very active, fairly active and lightly active minutes.

```
dailyActivity <- dailyActivity %>% mutate(TotalActiveMinutes=VeryActiveMinutes+
                                            FairlyActiveMinutes + LightlyActiveMinutes)
```

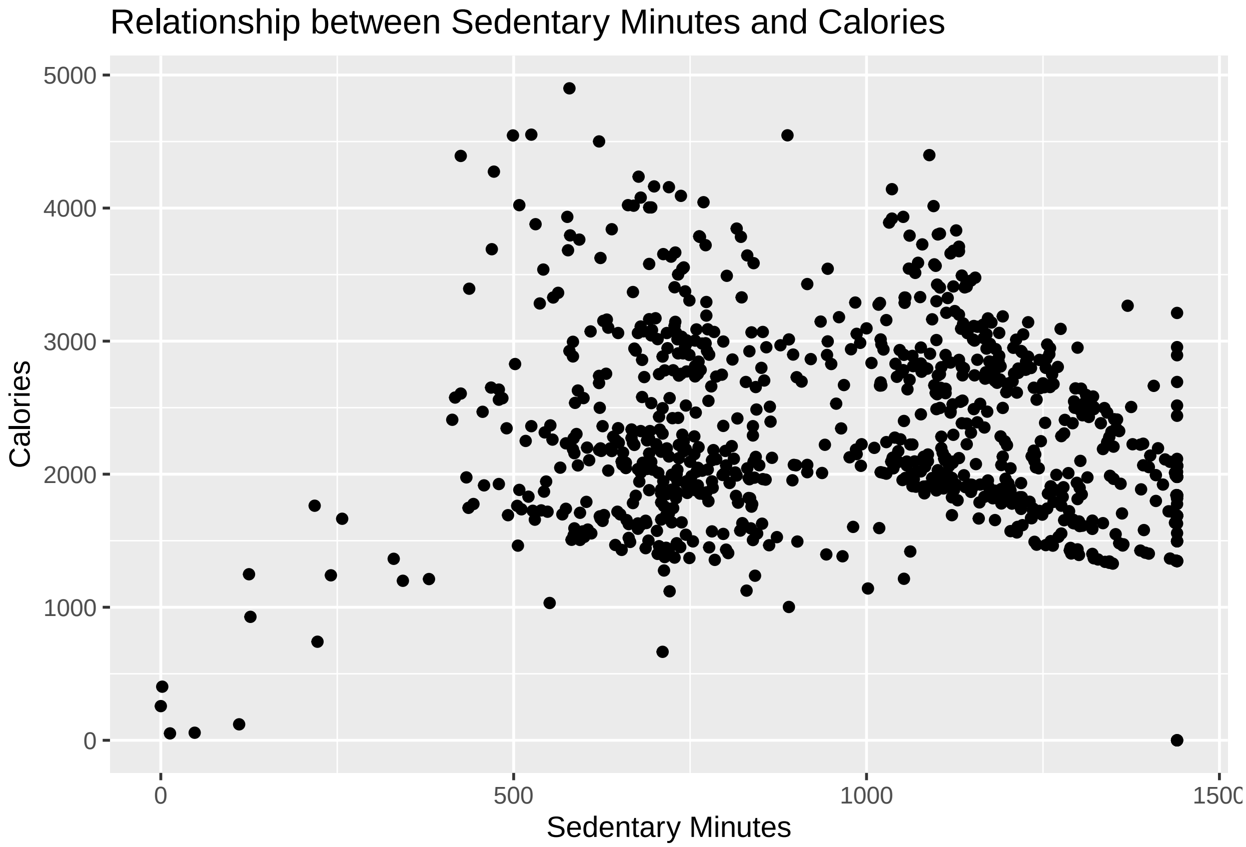
```
ggplot(data = dailyActivity)+
  geom_point(mapping = aes(x=TotalActiveMinutes, y=Calories))+
  labs(title = 'Relationship between Total Active Minutes and Calories Spent',
       x = 'Total Active Minutes',
       y = 'Calories')
```



There is a visible linear dependency between the total active minutes and calories spent. That is, as the total active minutes increased, the calories spent increased as well. However, there were records of calories spent with no active minutes.

Relationship between sedentary minutes and calories

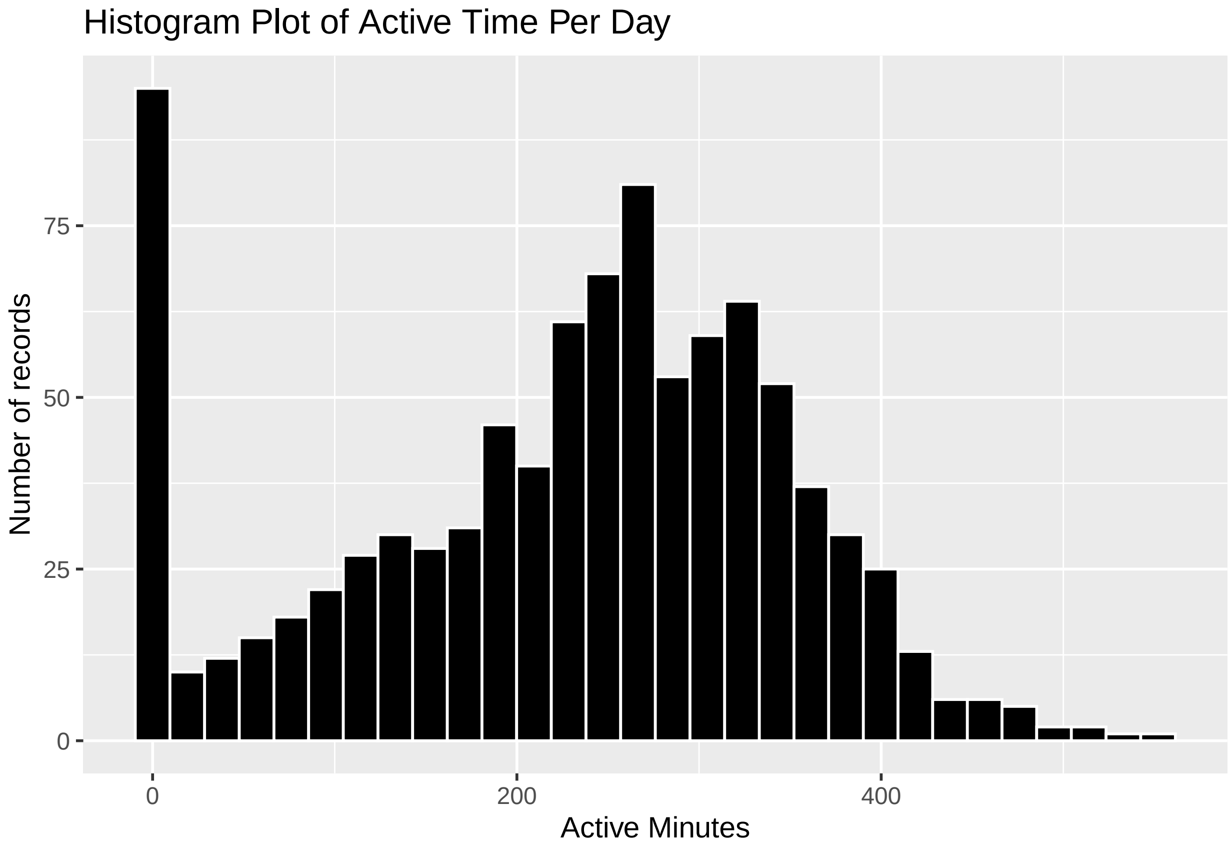
```
ggplot(data=dailyActivity)+
  geom_point(mapping = aes(x=SedentaryMinutes, y=Calories))+
  labs(title = 'Relationship between Sedentary Minutes and Calories',
       x = 'Sedentary Minutes',
       y = 'Calories')
```



How the users were active per day

```
ggplot(data=dailyActivity)+
  geom_histogram(mapping = aes(x=TotalActiveMinutes), color='white', fill='black')+
  labs(title = 'Histogram Plot of Active Time Per Day',
       x = 'Active Minutes',
       y = 'Number of records')
```

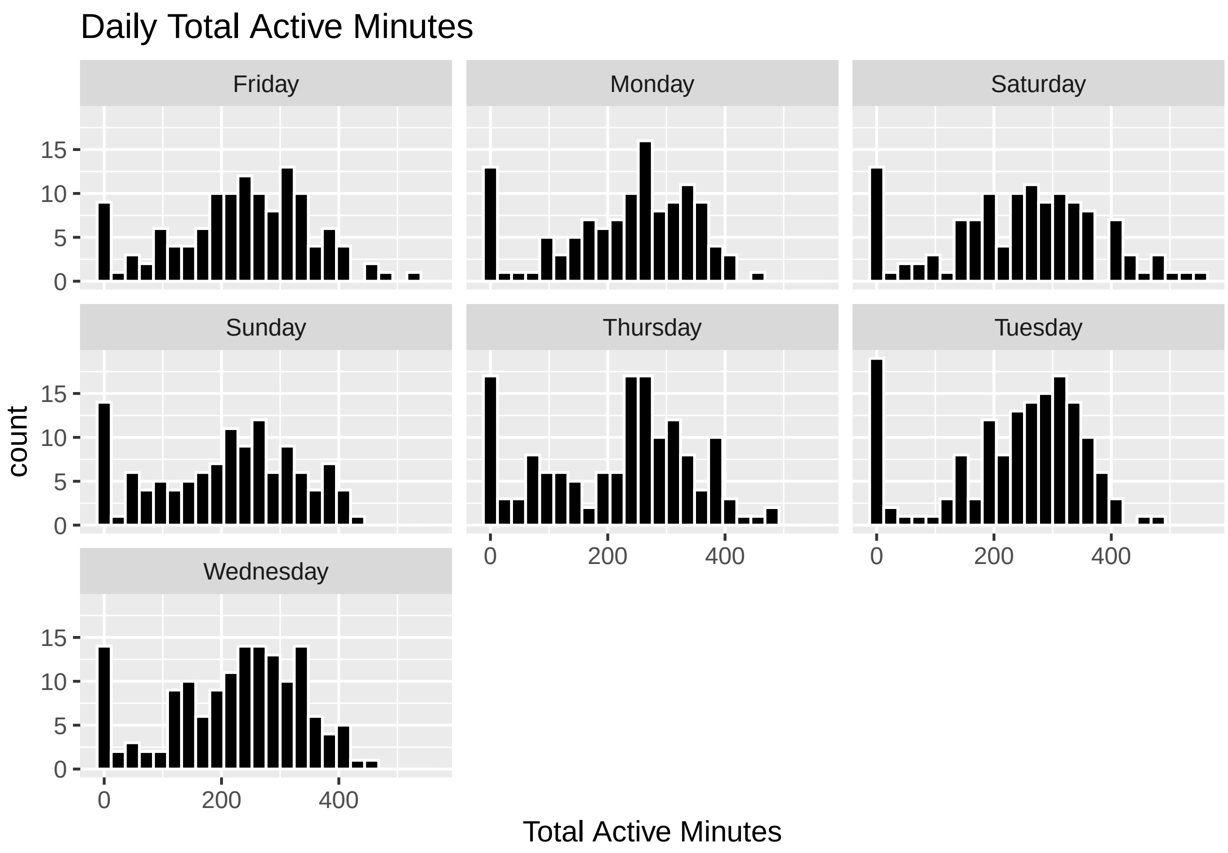
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Relationship between active minutes and day of the week

```
dailyActivity <- dailyActivity %>%
  mutate(DayOfWeek = weekdays(ActivityDate))
```

```
ggplot(data=dailyActivity)+
  geom_histogram(mapping = aes(x=TotalActiveMinutes), color='white', fill='black', bins=24)+
  facet_wrap(~DayOfWeek)+
  labs(title = 'Daily Total Active Minutes',
       x = 'Total Active Minutes')
```



At least 30 minutes of activity is recommended daily

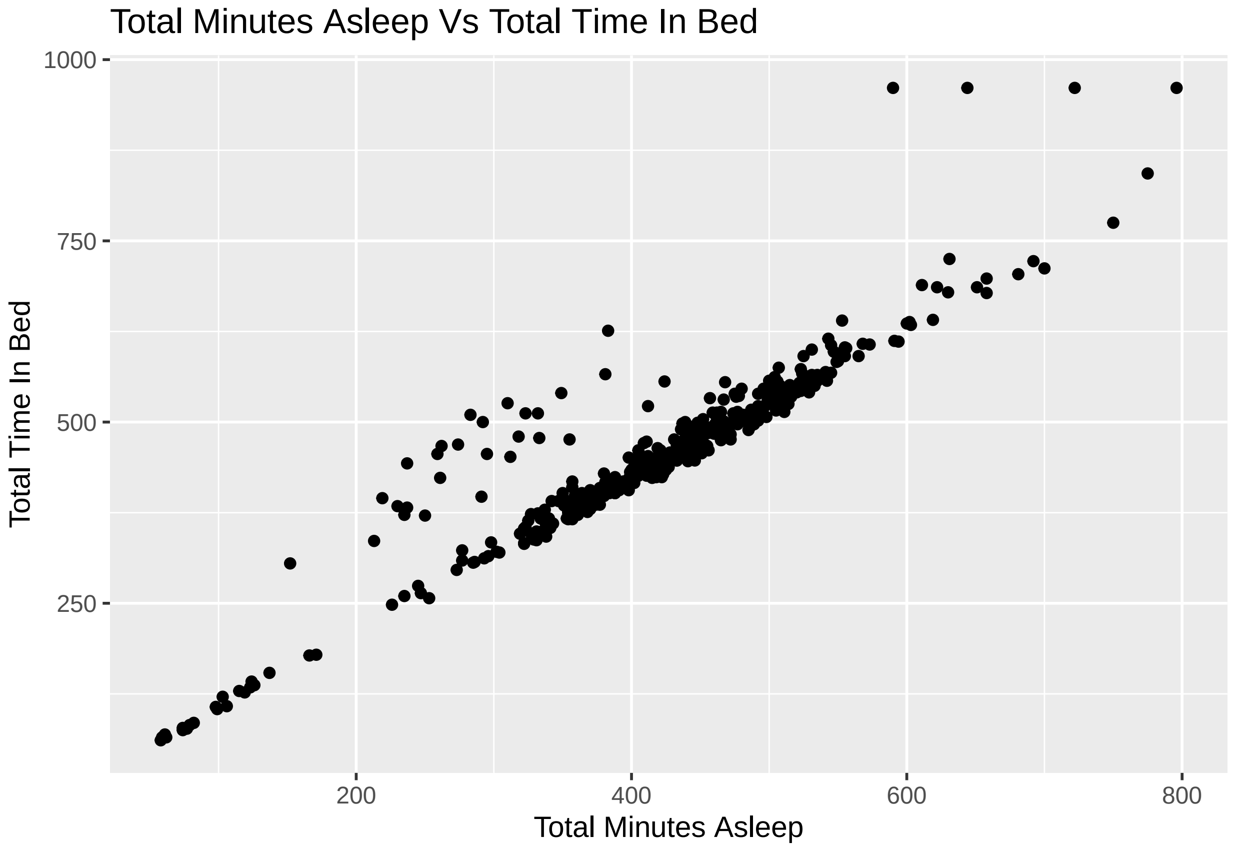
```
dailyActivity %>% group_by(DayOfWeek) %>%
  summarise(zero_activity_days = sum(TotalActiveMinutes==0),
            low_activity_days = sum(TotalActiveMinutes < 30),
            active_days = sum(TotalActiveMinutes >= 30),
            total_days = zero_activity_days+low_activity_days+active_days,
            active_days_percent = (active_days/total_days)*100)
```

```
## # A tibble: 7 x 6
##   DayOfWeek zero_activity_days low_activity_days active_days total_days active~1
##   <chr>                  <int>             <int>       <int>      <int>    <dbl>
## 1 Friday                     6                10         116        132     87.9
## 2 Monday                    11                14         106        131     80.9
## 3 Saturday                  11                13         111        135     82.2
## 4 Sunday                    12                14         107        133     80.5
## 5 Thursday                  14                18         129        161     80.1
## 6 Tuesday                   17                21         131        169     77.5
## 7 Wednesday                 12                15         135        162     83.3
## # ... with abbreviated variable name 1: active_days_percent
```

The table above shows that users were least active on Tuesdays while they were most active on Fridays.

Relationship between total minutes asleep and total time in bed

```
ggplot(data = sleepDay)+
  geom_point(mapping = aes(x = TotalMinutesAsleep, y = TotalTimeInBed))+
  labs(title = 'Total Minutes Asleep Vs Total Time In Bed',
       x = 'Total Minutes Asleep',
       y = 'Total Time In Bed')
```



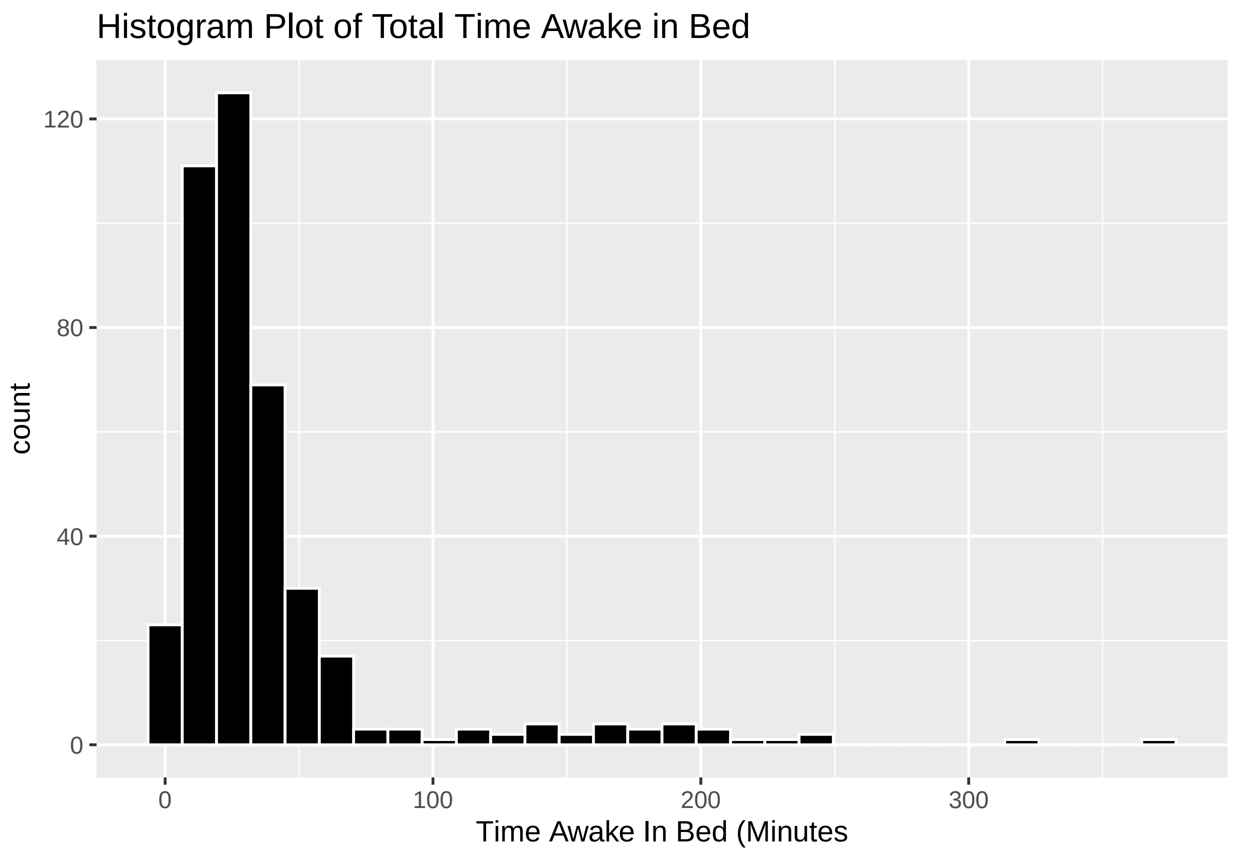
The total minutes asleep increased as the total time in bed increased.

Time in Bed Awake

```
sleepDay <- sleepDay %>% mutate(TimeInBedAwake = TotalTimeInBed-TotalMinutesAsleep)

ggplot(data=sleepDay)+
  geom_histogram(mapping= aes(x=TimeInBedAwake), color='white', fill='black')+
  labs(title = 'Histogram Plot of Total Time Awake in Bed',
       x = 'Time Awake In Bed (Minutes')
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



The plot shows that a few users were in bed for more than 200 minutes but were awake. However, most users spent less than 100 minutes awake while on bed.

Distribution of users' sleep time

420 minutes of sleep is recommended daily

```
sleepDay <- sleepDay %>%
  mutate(DayOfWeek = weekdays(SleepDate))
```

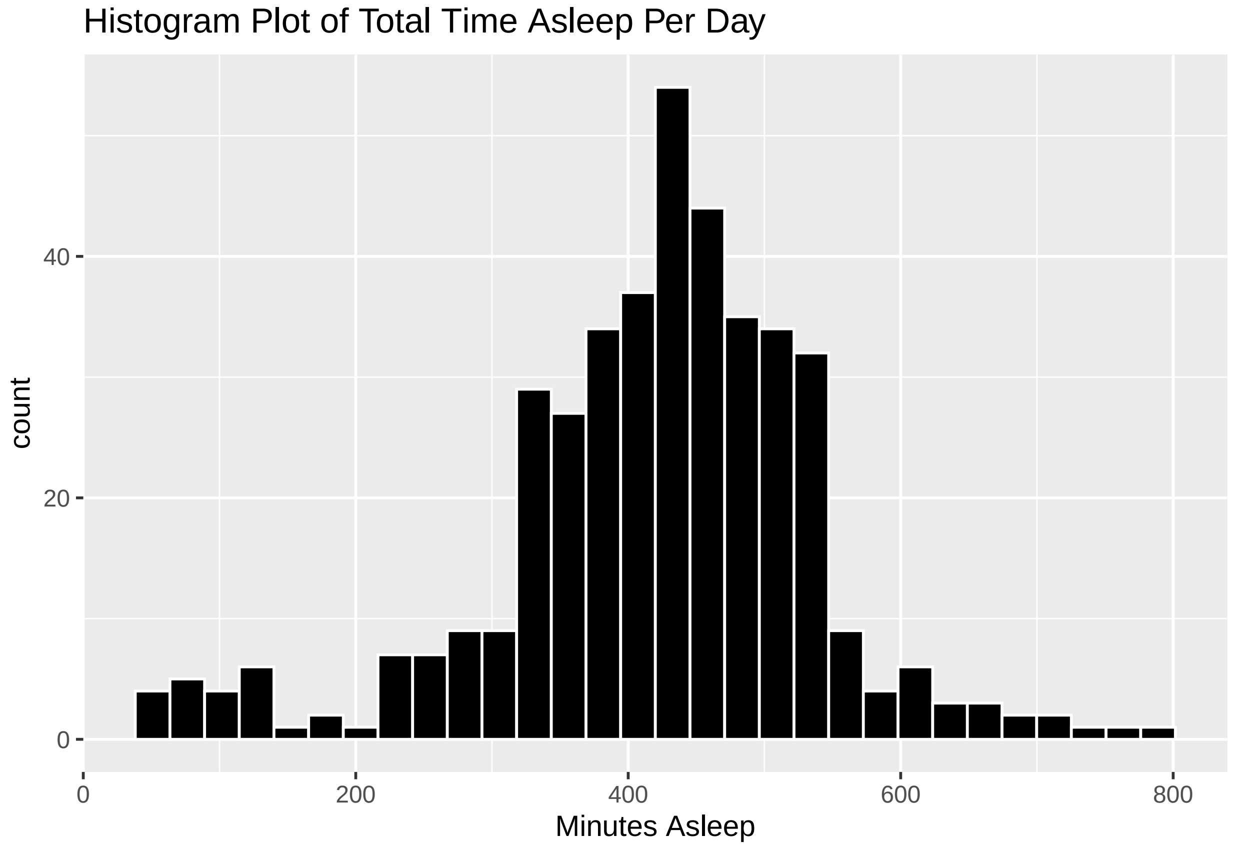
```
sleepDay %>% group_by(DayOfWeek) %>%
  summarise (Normal = sum(TotalMinutesAsleep >=420),
             Insufficient = sum(TotalMinutesAsleep < 420),
             Insufficient_percentage = (Insufficient/(Insufficient+Normal)*100))
```

```
## # A tibble: 7 x 4
##   DayOfWeek Normal Insufficient Insufficient_percentage
##   <chr>      <int>        <int>                   <dbl>
## 1 Friday        26           31                    54.4
## 2 Monday        27           20                    42.6
## 3 Saturday      31           27                    46.6
## 4 Sunday        37           18                    32.7
## 5 Thursday      35           30                    46.2
## 6 Tuesday       30           35                    53.8
## 7 Wednesday     45           21                    31.8
```

The table above shows that users slept for an insufficient amount of time mostly on Fridays and Tuesdays while Wednesdays and Sundays had the least insufficient sleep

```
ggplot(data=sleepDay)+
  geom_histogram(mapping= aes(x=TotalMinutesAsleep), color='white', fill='black')+
  labs(title = 'Histogram Plot of Total Time Asleep Per Day',
       x = 'Minutes Asleep')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Daily activity and sleep

The `dailyActivity` and `sleepDay` data were merged by unique user IDs and date. This was done to determine the relationship between active minutes, steps and sleep time.

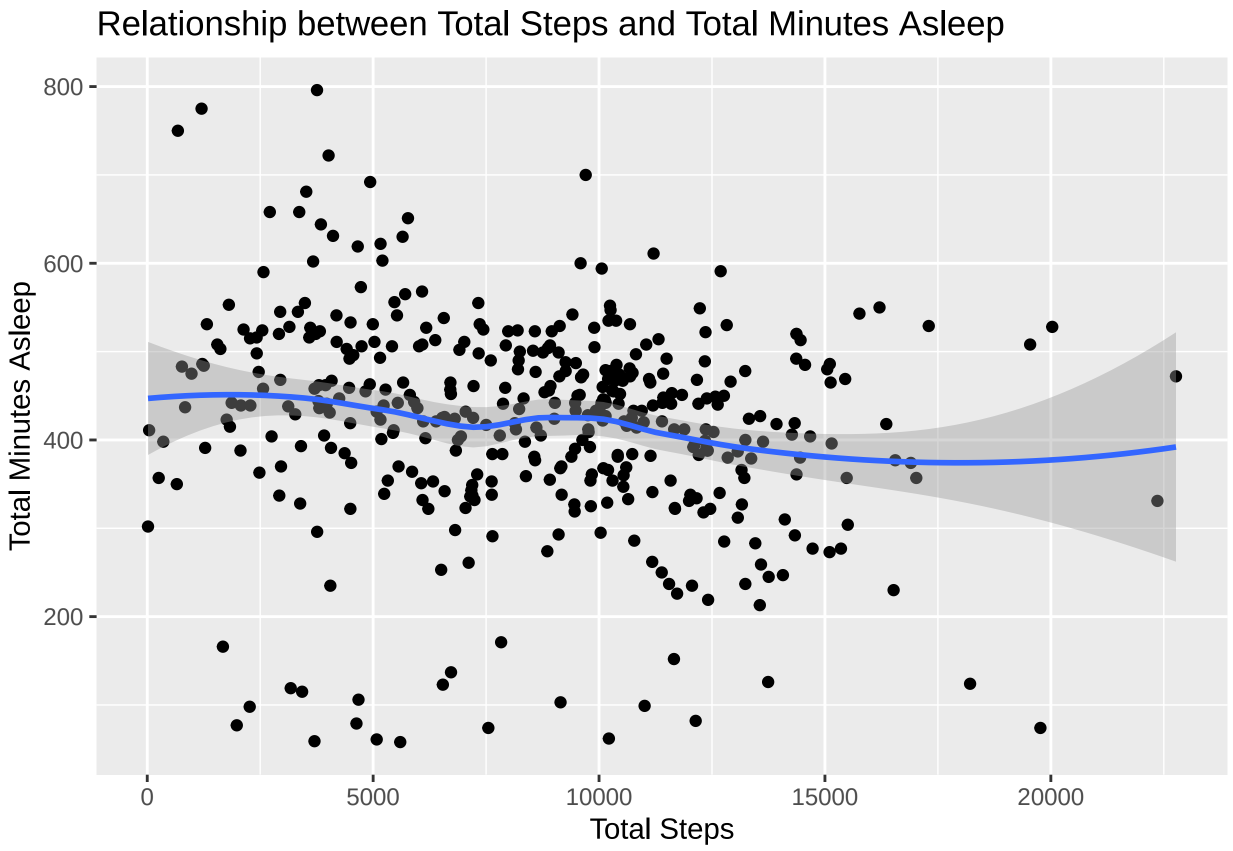
Merging the two datasets

```
dailyActivitySleep <- merge(dailyActivity, sleepDay,
by.x=c("Id","ActivityDate"), by.y=c("Id", "SleepDate"))
```

Relationship between total steps and total minutes asleep

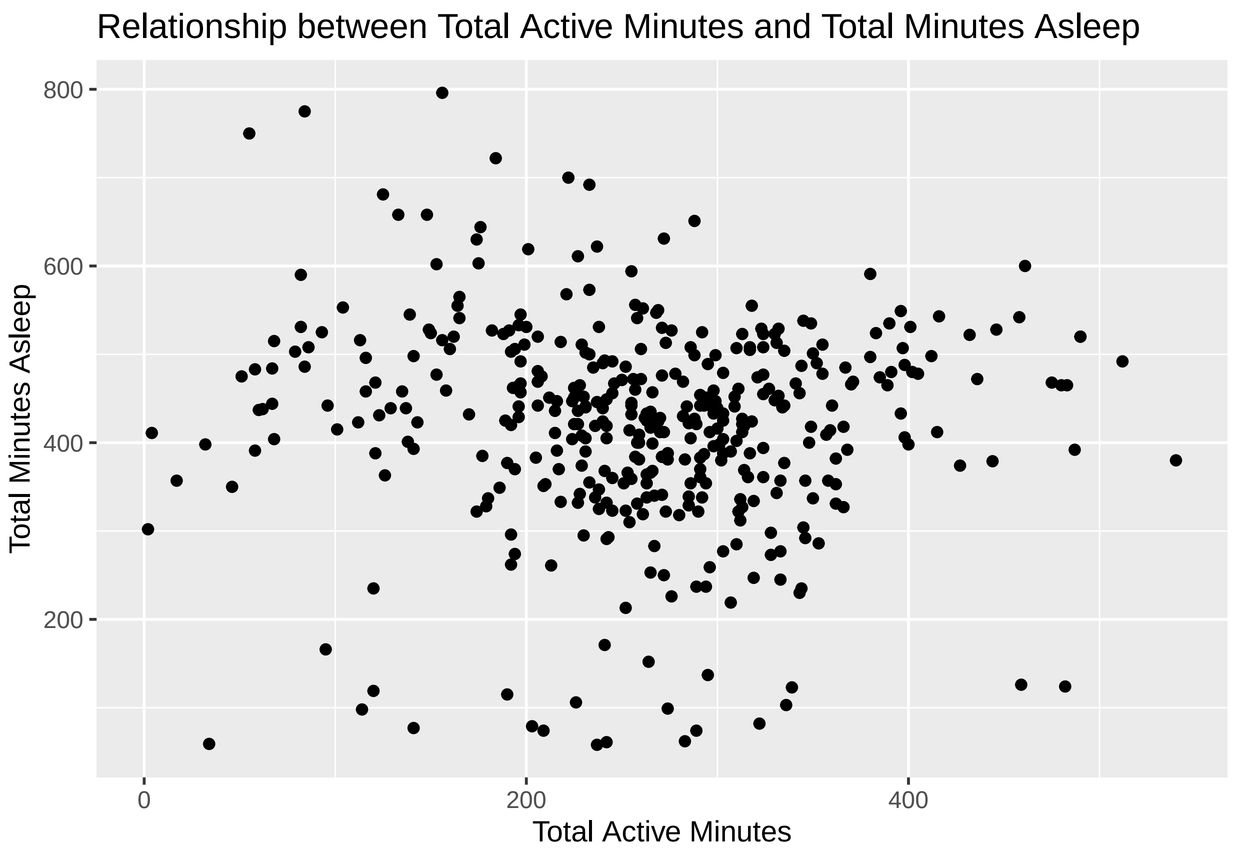
```
ggplot(data = dailyActivitySleep)+
  geom_point(mapping=aes(x=TotalSteps, y=TotalMinutesAsleep))+
  geom_smooth(mapping=aes(x=TotalSteps, y=TotalMinutesAsleep))+
  labs(title = 'Relationship between Total Steps and Total Minutes Asleep',
       x = 'Total Steps',
       y = 'Total Minutes Asleep')
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



The plot shows that there is no clear dependency between the two variables.
Relationship between total active minutes and total minutes asleep

```
ggplot(data = dailyActivitySleep)+
  geom_point(mapping=aes(x=TotalActiveMinutes, y=TotalMinutesAsleep))+
  labs(title = 'Relationship between Total Active Minutes and Total Minutes Asleep',
       x = 'Total Active Minutes',
       y = 'Total Minutes Asleep')
```



The plot shows that there is no clear relationship between the two variables. However, most users who were active between 200 and 300 minutes slept mostly between 300 and 500 minutes.

Days users met the recommended daily active minutes and sleep

```
dailyActivitySleep <- dailyActivitySleep %>%
  mutate(DayOfWeek = weekdays(ActivityDate))
dailyActivitySleep %>% group_by(DayOfWeek) %>%
  summarize(NumberOfDays = sum(TotalActiveMinutes>=0), CountActiveMinutesSleep = sum(TotalActiveMinutes
            PercentAMS = (CountActiveMinutesSleep/NumberOfDays)*100)
```

```
## # A tibble: 7 x 4
##   DayOfWeek NumberOfDays CountActiveMinutesSleep PercentAMS
##   <chr>            <int>                   <int>      <dbl>
## 1 Friday              57                      26       45.6
## 2 Monday              47                      27       57.4
## 3 Saturday            58                      31       53.4
## 4 Sunday              55                      37       67.3
## 5 Thursday            65                      35       53.8
## 6 Tuesday             65                      30       46.2
## 7 Wednesday           66                      45       68.2
```

Users mostly reached the recommended daily active minutes and sleep on Wednesday, then Sunday, while it occurred the least on Friday, then Tuesday.

Days users met the recommended daily steps and sleep

```
dailyActivitySleep %>% group_by(DayOfWeek) %>%
  summarize(NumberOfDays = sum(TotalSteps>=0), StepsAndSleep = sum(TotalSteps>=10000 & TotalMinutesAsle
            PercentSAS = (StepsAndSleep/NumberOfDays)*100)
```

```
## # A tibble: 7 x 4
##   DayOfWeek NumberOfDays StepsAndSleep PercentSAS
##   <chr>            <int>         <int>      <dbl>
## 1 Friday              57             6       10.5
## 2 Monday              47            10       21.3
## 3 Saturday            58            13       22.4
## 4 Sunday              55            12       21.8
## 5 Thursday            65            13       20
## 6 Tuesday             65            12       18.5
## 7 Wednesday           66            14       21.2
```

Users mostly reached the recommended daily steps and sleep on Saturday, then Sunday, while it occurred the least on Friday, then Tuesday.

Days users met the recommended daily active minutes, steps and sleep

```
dailyActivitySleep %>% group_by(DayOfWeek) %>%
  summarize(NumberOfDays = sum(TotalSteps>=0), ActiveMinutesStepsAndSleep = sum(TotalSteps>=10000 & Tota
            PercentAMSAS = (ActiveMinutesStepsAndSleep/NumberOfDays)*100)
```

```
## # A tibble: 7 x 4
##   DayOfWeek NumberOfDays ActiveMinutesStepsAndSleep PercentAMSAS
##   <chr>            <int>                      <int>        <dbl>
## 1 Friday              57                          6         10.5
## 2 Monday              47                         10         21.3
## 3 Saturday            58                         13         22.4
## 4 Sunday              55                         12         21.8
## 5 Thursday            65                         13         20
## 6 Tuesday             65                         12         18.5
## 7 Wednesday           66                         14         21.2
```

Users mostly reached the recommended daily active minutes, steps and sleep on Saturday, then Sunday, while it occurred the least on Friday, then Tuesday.

Weight log analysis

Summary of weight data

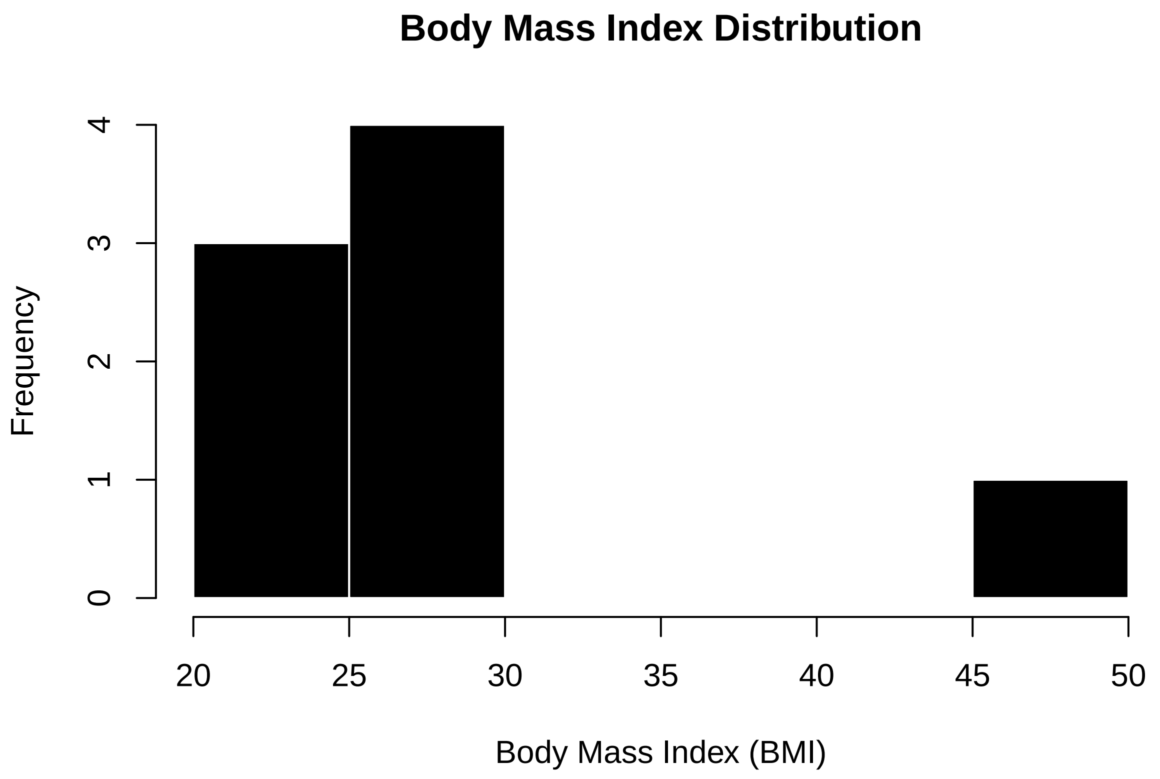
```
meanWeight <- weightLog %>%
  group_by(Id) %>%
  summarise(meanBMI = mean(BMI), meanKG = mean(WeightKg))
```

```
as_tibble(meanWeight)
```

```
## # A tibble: 8 x 3
##           Id meanBMI meanKG
##        <dbl>   <dbl>  <dbl>
## 1 1503960366    22.6   52.6
## 2 1927972279    47.5  134.
## 3 2873212765    21.6   57
## 4 4319703577    27.4   72.4
## 5 4558609924    27.2   69.6
## 6 5577150313    28     90.7
## 7 6962181067    24.0   61.6
## 8 8877689391    25.5   85.1
```

BMI Distribution

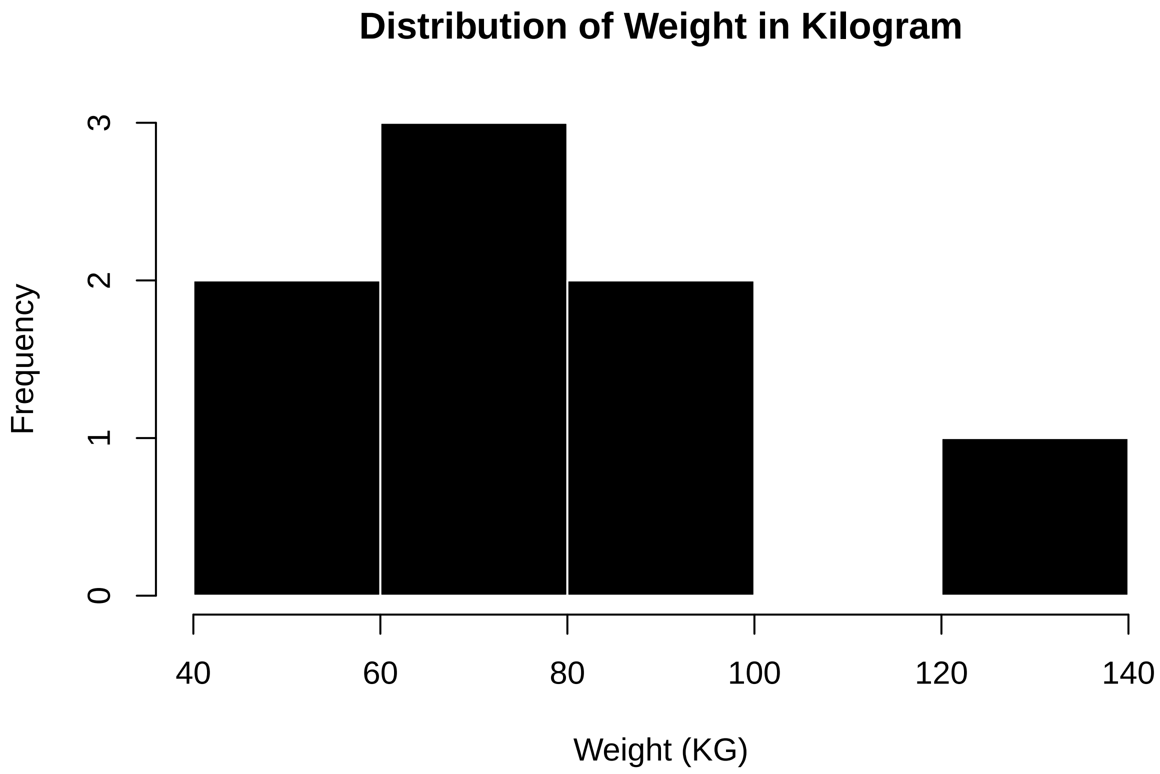
```
hist(meanWeight$meanBMI,
     main='Body Mass Index Distribution',
     xlab='Body Mass Index (BMI)',
     col =  'black',
     border = 'white')
```



Only 8 users recorded their weight data. Of these, 3 are in the BMI range of 20-25 which is considered as healthy weight (18.5-24.9), 4 users are overweight (25-29.9), and one user is in the range of 45-50 which is considered obese (>30).

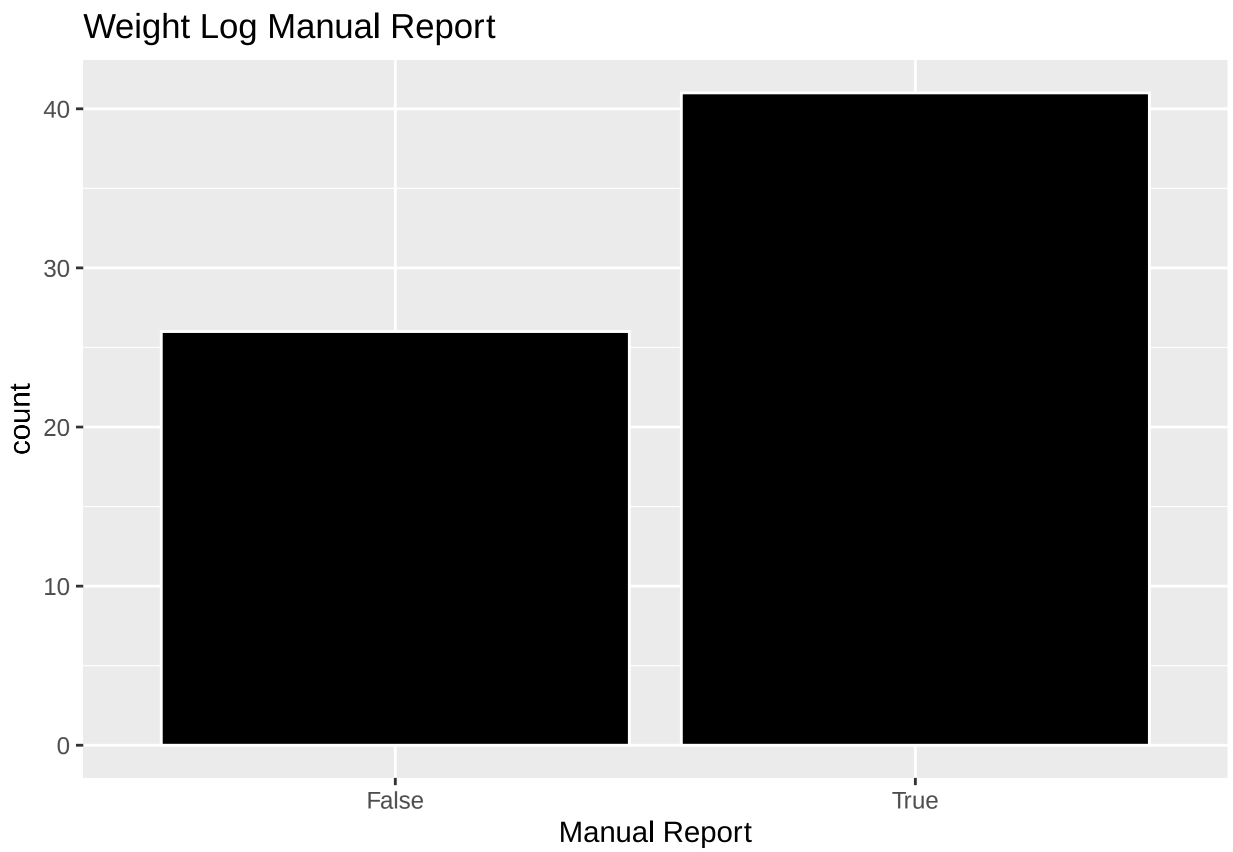
Distribution of weight in kilogram(KG)

```
hist(meanWeight$meanKG,
     main = 'Distribution of Weight in Kilogram',
     xlab = 'Weight (KG)',
     col =  'black',
     border = 'white')
```



How the weight log was reported

```
ggplot(data=weightLog)+
  geom_bar(mapping = aes(x = IsManualReport), color='white', fill='black')+
  labs(title = 'Weight Log Manual Report',
       x = 'Manual Report')
```



41 out of 67 weight records were manually recorded.

ACT

Summary of key findings and recommendations

1. Analysis of the data revealed that users were least and less active on Tuesdays and Thursdays, respectively. Some users also walked an average of fewer than 10000 steps on average. I suggest a timed notification for each user encouraging them to increase their daily activity to meet the recommendations of at least 30 minutes of moderate physical activity and 10000 steps every day. Reward points should be given to users who reach their daily goals. Such points can be used to unlock or get discounts on premium features on Bellabeat app, such as workout routines and customized meal plans.

2. The analysis carried out also showed that users tend to have insufficient sleep, especially on Fridays and Tuesdays. I recommend allowing users to set their preferred bedtime with the option of a reminder 30 minutes before that time and to set a preferred time to disable specific notifications. For users whose data show that they often spend more than 30 minutes to 1 hour in bed before sleeping, the Bellabeat app could provide recommendations on good bedtime routines such as light meals, warm baths, listening to music, or journaling.

3. Based on the insights from the analysis, digital marketing for Bellabeat products can be carried out on days were users were most likely to use them to track their activities in order to meet their goals. For example, users mostly met the recommended daily goals of 10000 steps, 30 active minutes and 420 minutes of sleep on Saturdays and Sundays, while it was less likely to occur on Fridays and Tuesdays.

4. Only a few users (8 of 33) recorded their weight data, and 41 of 67 records were recorded manually. I recommend adding a feature on the Bellabeat app that allows users on a weight loss journey to connect, to share their success stories, and for users with similar goals to team up.

REFERENCES

1. How Much Sleep Do I Need? Sleep and Sleep Disorders. Centers for Disease Control and Prevention, 14 Sept. 2022. https://www.cdc.gov/sleep/about_sleep/how_much_sleep.html

2. How many steps should people take per day? Medical News Today, 20 Jan. 2021. https://www.medicalnewstoday.com/articles/how-many-steps-should-you-take-a-day

3. How Much Physical Activity Do Adults Need? Centers for Disease Control and Prevention, 2 June 2022. https://www.cdc.gov/physicalactivity/basics/adults/index.htm

4. About Adult BMI. Centers for Disease Control and Prevention, 3 June 2022. https://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/index.html